Using Graphs to Visualise and Detect Anomalies in Access Control System Transactions

Michael Davis

Centre for Secure Information Technologies (CSIT),
The Institute of Electronics, Communications and Information Technology (ECIT),
Queen's University, Belfast (QUB),
Northern Ireland Science Park, Belfast, BT3 9DT, United Kingdom
mdavis05@qub.ac.uk

Abstract. Physical Access Control Systems are commonly used to secure doors in buildings such as airports, hospitals, government buildings and offices. These systems are designed primarily to provide an authentication mechanism, but they also log each door access as a transaction in a database. Unsupervised learning techniques can be used to detect inconsistencies or anomalies in the mobility data, such as a cloned or forged Access Badge, or unusual behaviour by staff members.

In this paper, we present an overview of our method of inferring directed graphs to represent a physical building network and the flows of mobility within it. We demonstrate how the graphs can be used for Visual Data Exploration, and outline how to apply algorithms based on Information Theory to the graph data in order to detect inconsistent or abnormal behaviour.

Keywords: graph visualisation, graph mining, anomaly detection, surveillance

1 Introduction

Physical Access Control Systems (ACS) are mandatory in secure facilities such as airports and power stations, and ubiquitous in many other types of building, such as hospitals, government departments and offices. ACS users present their credentials (typically an Access Badge) to a door sensor. The system validates the user's identity and compares it to the Access Control Policy to decide whether access should be granted or denied. The policy can restrict access to certain zones or times. In a large installation, policy rules are complex and often contain errors.

The current generation of Access Control Systems can detect *suspicious events*, such as the use of an Access Badge which has been reported stolen, or an attempt to physically force a door. They cannot detect *suspicious patterns*—transactions which are innocent in themselves but anomalous when considered in context: an airport technician who regularly hangs around in the baggage handling area, or a clerk who is spending an unusually long time on their own in the cash room.

When an Access Badge is presented to a sensor, a transaction is generated and stored in a database. Analysis of this data is typically limited to reports, alerts or rules created by domain experts [6]. In our research, we will use unsupervised learning techniques to detect *inconsistencies*—*e.g.* someone who appears to move from one area to another in an infeasibly short amount of time—and *anomalies* in mobility patterns. Inconsistencies may represent an intrusion using a cloned Access Badge, while anomalies may indicate suspicious behaviour or rare events, warranting further investigation in either case.

There has been relatively little published research in the area of physical Access Control Systems, but there have been some attempts to apply lessons from information access control. [6] proposes the use of Role-Based Access Control (RBAC) for building security and suggests dynamically changing the access policy to restrict where a "suspicious" user can go. The definition of "suspicious" is left to a human expert. [7] compares an Expert System (user-defined rules) with event classification (C4.5 decision trees) and outlier detection (LOF). However, the dataset for this study is extremely small (only one sensor and five users).

ACS transactions are analysed to detect four kinds of suspicious patterns in [2]. There is a separate algorithm for detecting each type of anomaly, using a simple parameterised approach. Rather than have four separate algorithms, we propose to use a single graph-theoretic algorithm which can detect both structural anomalies (unusual paths through the building) and numeric anomalies (unusual timing data).

[1] uses Association Rule Mining (Apriori) to detect ACS policy misconfigurations and predict required changes before the policy interferes with legitimate accesses. We suggest that a graph-based approach using frequent subgraph mining may be better able to process a dataset which is larger and more complex than the one used in this study.

Graphs have been used for Visual Analytics in Internet Intrusion Detection Systems [9], significantly improving on previous visualisation approaches, which used simple statistical graphs. A similar Visual Analytic approach can be applied to the real-time detection of abnormal behaviour in secure buildings.

We propose to develop our work by using our graph model to detect inconsistent and abnormal behaviour. A number of algorithms for detecting structural anomalies in graphs have been presented, including Subdue [10], Autopart [3] and GBAD [4]. Characteristically, these algorithms work on unweighted graphs with discrete vertex and edge labels, *i.e.*, they cannot handle numeric attributes in the graph. Recently [8, 11, 12], there have been some efforts towards frequent subgraph detection on weighted graphs and graphs with numeric labels on vertices and edges. These approaches do not specifically consider the problem of anomaly detection, however. In [5], the GBAD-P algorithm is extended to handle anomaly detection in numeric attributes, but assumes that the numeric values follow a Gaussian distribution.

In the next section, we describe our use of graphs for visualising the physical building network and the flows of mobility through it. We present our conclusions

to date in Sect. 3, including our proposal to apply graph-theoretic approaches to anomaly detection in our future work.

2 Graph Visualisation of Transaction Data

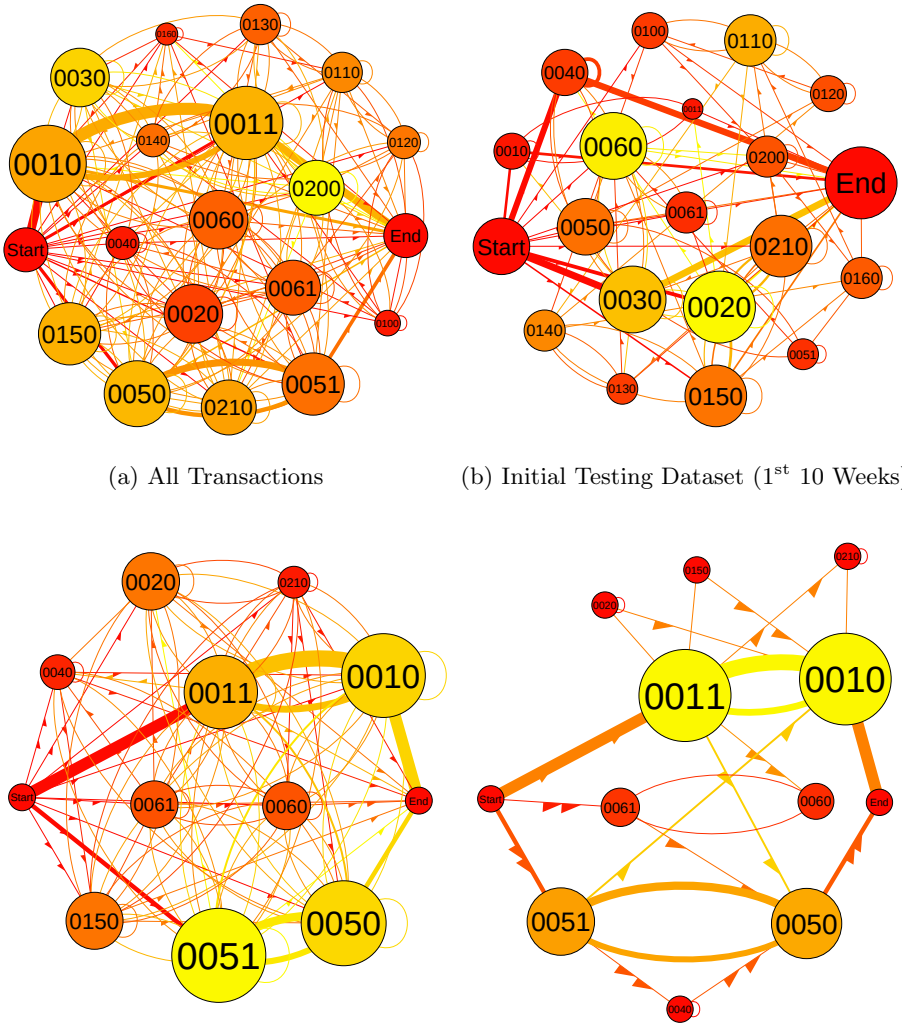
While there is some previous work which analyses ACS data for physical security applications, the novelty of our approach is to represent the topology of the building and the mobility flows within it as a weighted, directed graph $G = (V, E)$. Each vertex $v \in V$ represents a door sensor. Some doors have a sensor on both sides, so each physical door is represented by one or two vertices. The graph edges $e \in E$ represent movements between sensors, with weight $\mathcal{W}(e)$ equal to the number of movements along that edge. The positioning of the sensors and the temporal ordering of the transactions imply a direction of travel for each edge.

The graphs inferred from the ACS transactions are shown in Fig. 1. The size of each vertex is proportional to its Degree, the number of transactions which take place at that sensor. The colour of the vertex represents its Betweenness Centrality (red is low, yellow is high). These measures allow us to quickly identify main entrances and thoroughfares (0011–0010 and 0051–0050) and to distinguish them from "dead-end" rooms which were later identified as Server Rooms (0020, 0150, 0210) and the Document Room (0040). The weight of each edge represents the number of traversals of each pathway, *i.e.* it is proportional to the frequency of each path segment in the dataset.

Visual Data Exploration can be seen as a hypothesis generation process [9]. It is particularly suitable when the data is noisy and difficult to understand, as was the case in Fig. 1a. Through an iterative process of creating and visualising network graphs, we discovered that the patterns during the first 10 weeks of the dataset (Fig. 1b) were substantially different to those in the rest of the dataset (Fig. 1c). We hypothesised that the early part of the dataset represented a test of the system and identified the date that the system went live. We were later able to confirm this hypothesis with the owner of the data.

Next, we investigated the effect of missing transactions caused by the common practice of "tailgating" (following another person through a door without presenting a credential). The effect is to create edges in the graph which should not be there. For example, if a user must pass sensors A and B in order to get to C, but fails to swipe their Access Badge at B, a spurious edge will be created from A→C. We wish to detect and remove such edges from the graph.

We note two characteristics of paths created by tailgating. First, tailgating will always be represented as a shortcut, so there must exist a longer path connecting the end vertices. Second, we expect that valid paths will be followed more often than invalid paths, so the weight of edges created by tailgating will be lower than edges along valid paths. Our algorithm begins by considering each edge in the graph in turn, starting with the lowest-weighted edges. For each edge tuple (v_1, v_2), we search for all simple paths which satisfy $\{v_1, \ldots, v_2 : (v_i, v_{i+1}) \in E\}$. Once we have a set of candidate paths, we compare their weights. For the pur-



(a) All Transactions

(b) Initial Testing Dataset (1st 10 Weeks)

(c) Live System Dataset

(d) Filtered Live System Dataset

Fig. 1: **Directed Network Graphs:** Vertices represent Access Control sensors. Straight (directed) edges represent a movement from one vertex to another. Curved edges represent mutual flows of mobility between two vertices. The weight of each edge represents the number of traversals of each pathway.

poses of this comparison, we consider only the lowest weighted edge found along each path. If the edge under consideration is of lower weight than any of the other (longer) paths, the edge is considered spurious and is deleted from the graph.

The result of applying this algorithm to Fig. 1c is shown in Fig. 1d. Our algorithm has successfully removed all the spurious edges caused by tailgating, and the pattern of valid movements through the building can now be clearly seen.

3 Conclusions and Future Work

We have presented an overview of our algorithms to infer a directed network graph from an ACS database and to correct the graph for missing transactions caused by tailgating. The graph represents an accurate topological map of the building network and the valid flows of mobility which take place within it.

We demonstrated the use of graphs for Visual Data Exploration to clean a dataset of mobility transactions prior to further analysis. A similar graph-based approach can also be used for Visual Analytics, for real-time monitoring of building security, to highlight possible security weaknesses and threats and inform access policies.

The algorithm outlined in the previous section is a simple heuristic. Having identified some of the important characteristics of the mobility graphs, we intend to develop more formal methods for evaluating them. We propose that tailgating behaviour could be detected by searching for frequent subgraphs and applying an information theoretic algorithm to evaluate whether paths through the network are normal or anomalous. We also intend to consider how probabilistic graph approaches (*cf.* GBAD-P [4]) could be used to model the flow of the access transactions and the uncertainties in the network.

In our future work, we will extend our model to encompass labelled graphs: $G = (V, E, L^V, L^E, \mathcal{L}_V, \mathcal{L}_E)$, where L^V is a set of vertex labels, L^E is a set of edge labels, and $\mathcal{L}_V$ and $\mathcal{L}_E$ are label-to-value mapping functions. While V represents the entities in the graph and E represents the relationships between entities, L^V and L^E represent the attributes of the entities. We will allow that L^V and L^E contain numeric (continuous) attributes and will present an algorithm which can detect both structural and numeric anomalies in graph data.

The work presented in this paper inferred a flat, static graph. For visualising and manipulating larger datsets, it will be necessary to extend our algorithms to infer dynamic and hierarchical graphs. A hierarchical graph can cluster vertices together based on considerations such as proximity or being along the same path. Vertices can thus be expanded or collapsed for ease of viewing. We also propose to infer dynamic graphs, so that mobility flows can be viewed as an animated sequence. Dynamic graphs effectively provide a higher-dimensional view of the data which will enable more complex anomaly detection algorithms.

Although we have focussed on security applications, there are other uses for visualising mobility networks. We are involved in a health promotion project

where users touch sensors as they exercise in a local park. Graph Visualisation can be used to identify the most popular pathways taken through the park and identify changes in behaviour as a result of participation in the health promotion scheme. These techniques could also be applied to the visualisation of mobility data generated by GPS transceivers in the current generation of mobile phones.

References

1. Bauer, L., Garriss, S., Reiter, M.K.: Detecting and resolving policy misconfigurations in access-control systems. In: SACMAT'08: 13th ACM Symposium on Access Control Models and Technologies. pp. 185–194. ACM SIGSAC, ACM (2008)
2. Biuk-Aghai, R.P., Si, Y.W., Fong, S., Yan, P.F.: Security in physical environments: Algorithms and system for automated detection of suspicious activity. In: BI'10: Proceedings of the Workshop on Behavior Informatics 2010. Springer-Verlag (2010)
3. Chakrabarti, D.: Autopart: Parameter-free graph partitioning and outlier detection. In: PKDD. pp. 112–124 (2004)
4. Eberle, W., Holder, L.: Mining for insider threats in business transactions and processes. In: Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on. pp. 163–170 (march 2009)
5. Eberle, W., Holder, L.B.: Discovering anomalies to multiple normative patterns in structural and numeric data. In: FLAIRS Conference (2009)
6. Fernandez, E.B., Ballesteros, J., Desouza-Doucet, A.C., Larrondo-Petrie, M.M.: Security patterns for physical access control systems. In: Data and Applications Security XXI. LNCS, vol. 4602, pp. 259–274. Springer-Verlag, Berlin (2007)
7. Gams, M., Tusar, T.: Intelligent high-security access control. Informatica 31(4), 469–477 (2007)
8. Jiang, C., Coenen, F., Zito, M.: Frequent sub-graph mining on edge weighted graphs. In: Proceedings of the 12th international conference on Data warehousing and knowledge discovery. pp. 77–88. DaWaK'10, Springer-Verlag, Berlin, Heidelberg (2010)
9. Mansman, F., Meier, L., Keim, D.A.: Visualization of host behavior for network security. In: VIZSEC 2007. pp. 187–202. Mathematics and Visualization, Springer-Verlag KG, Berlin (2008), 4th International Workshop on Computer Security, Sacramento, CA, Oct 29, 2007
10. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 631–636. KDD '03, ACM, New York, NY, USA (2003)
11. Perez, G., Olmos, I., Gonzalez, J.A.: Subgraph isomorphism detection with support for continuous labels. In: The 23rd International FLAIRS Conference (2010)
12. Romero, O.E., Gonzalez, J.A., Holder, L.B.: Handling of numeric ranges for graph-based knowledge discovery. In: The 23rd International FLAIRS Conference (2010)